

ECO-Mem: An Event-Centric Organization Framework for Visibility-Controlled Long-Term Memory

Xing Pei¹, Dandan Song^{1*}, Zhijing Wu¹, Yuhang Tian¹, Jun Yang¹
Yitian Ma¹, Yading Li¹, Xudong Li¹, Huipeng Ma¹, Yan Xu¹

¹School of Computer Science and Technology, Beijing Institute of Technology, China
{peixing, sdd}@bit.edu.cn

Abstract

Large language models have achieved strong single-turn language understanding and generation abilities, but long-term agent applications require models to handle dynamic interaction history beyond an isolated inference call. Existing agent memory systems address this need through external storage and retrieval and have achieved promising results. However, many existing frameworks rely on full storage or highly structured memory representations, which can increase cost and noise while disrupting event coherence. Inspired by cognitive views of event organization and adaptive forgetting, we introduce ECO-Mem, an event-centric organization framework for visibility-controlled long-term memory. ECO-Mem segments dialogue into source-linked event episodes and connects them through typed event relations, representing memory as linked events rather than isolated facts. It implements forgetting as non-destructive visibility control instead of destructive deletion. At query time, it traverses the event-relation graph to identify relevant hidden support and reactivates it only when the source evidence confirms the query context. Experiments on LoCoMo and RealTalk validate ECO-Mem’s effectiveness and efficiency for long-term agent memory.

1 Introduction

Large language models (LLMs) have advanced rapidly as general-purpose text models. By internalizing large-scale textual knowledge, they acquire broad parametric knowledge and striking one-shot or few-shot question-answering ability (Brown et al., 2020). Having this capability alone is not sufficient for serving as a long-term human assistant. A single answer generator can respond to the current prompt, but a persistent assistant must track changing interaction history, distinguish current information from outdated infor-

mation, retain commitments across sessions, and reuse earlier evidence when later questions depend on it. These requirements are not simply solved by larger context windows.

To address this gap, agent frameworks extend LLMs beyond static text prediction and few-shot question-answering ability. They enable a model to interact with external environments over multiple steps and adapt its behavior from experience (Park et al., 2023; Yao et al., 2023; Shinn et al., 2023). In cognitive accounts, memory is a core mechanism for processing, maintaining, and updating information under changing task demands (Baddeley, 1992); accordingly, memory is also a core module in agent frameworks. Agent memory can be viewed as an extraction–processing–reuse lifecycle: it extracts information from the interaction stream, stores and reorganizes that information, and reuses relevant evidence during later generation (Zhang et al., 2025; Hu et al., 2025; Fang et al., 2026). Existing systems instantiate this lifecycle through profile-based long-term memory, policy-driven memory management, temporal or note-based graph memory, and cost-aware consolidation (Zhong et al., 2024; Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025; Rasmussen et al., 2025; Fang et al., 2026).

These systems have made memory more controllable than raw dialogue histories and have shown clear benefits for long-term interaction. However, current extraction–processing–reuse memory designs still struggle when long interaction histories must be reused for answer generation. Long contexts are vulnerable to lost-in-the-middle behavior, causing key information to be missed even when it is present (Liu et al., 2024). Full or over-exposed memory introduces a separate problem: it increases retrieval noise and prompt cost. Structured storage can reduce exposure, but it may also cut dialogue evidence into isolated units, separating preference updates, correc-

*Corresponding author.

tions, and multi-turn events from their surrounding context. Cost-aware memory frameworks further tend to optimize one-time construction or storage costs, while persistent agents repeatedly pay QA-time costs over the same memory. These problems are not yet systematically addressed in a unified long-term agent memory design.

Studies in cognitive science suggest a different view of memory. Continuous experience is organized around events, which provides a natural unit for preserving coherent episodes (Zacks et al., 2007). Memory is structured through associations among episodes rather than isolated records, allowing later experiences to develop, revise, or continue earlier ones. Biological forgetting is not simply a way to reduce storage burden: it can reduce interference and improve flexibility (Nørby, 2015; Richards and Frankland, 2017). Classical memory studies further distinguish availability from accessibility, where traces may remain stored but fail to be retrieved until appropriate reactivation conditions arise (Tulving and Pearlstone, 1966), and the encoding-specificity principle suggests that retrieval is more effective when information available at test time overlaps with the original episode (Tulving and Thomson, 1973).

Motivated by these observations, we propose ECO-Mem: *An Event-Centric Organization Framework for Visibility-Controlled Long-Term Memory*. ECO-Mem has three components. First, it segments dialogue streams into source-linked event episodes, making events the basic units for storage, retrieval, and pruning while keeping them traceable to the original conversational evidence. Second, it reorganizes episodes into a typed event graph, where event-flow, state-chain, correction-chain, and continuation-chain edges preserve temporal development, state updates, revisions, and cross-episode continuity. Third, it applies visibility-controlled forgetting, which hides redundant or low-utility events from routine retrieval without erasing their source evidence, archive pointers, or graph relations. At query time, ECO-Mem uses relation-guided graph traversal to find relevant event paths and reactivates hidden support only when the archived source evidence confirms the query context, producing compact evidence for generation without forcing every stored detail into the prompt.

Our contributions are as follows:

- We highlight a limitation shared by several

existing long-term agent memory designs: full memory reuse can increase noise and QA-time cost, while structured storage can split dialogue evidence into isolated units and make coherent events harder to recover.

- We propose ECO-Mem, an event-centric memory framework that combines ECO event segmentation, typed event relations, visibility-controlled forgetting, and a source-confirmed step for visibility-controlled reactivation during compact answer-time reuse.
- We conduct comprehensive experiments across different datasets and settings, providing empirical evidence for the value of ECO-Mem as an effective and efficient long-term agent memory framework.

2 Related Work

Long-term memory for LLM agents. MemoryBank stores user interaction history as long-term memory for personalized responses (Zhong et al., 2024). MemGPT treats memory management as operating-system-like control over context and external storage (Packer et al., 2023). Mem0 builds scalable agent memory for production settings (Chhikara et al., 2025). A-MEM formulates memory construction and use as agentic operations for LLM agents (Xu et al., 2025). Generative Agents maintains a retrieved memory stream, while MIRIX coordinates multiple structured memory types (Park et al., 2023; Wang and Chen, 2025). LightMem designs lightweight memory-augmented generation through memory consolidation (Fang et al., 2026). Recent surveys summarize agent memory around extraction, organization, and utilization stages (Zhang et al., 2025).

Retrieval, graph, and temporal memory. Retrieval-augmented generation retrieves external passages before generation (Lewis et al., 2020). GraphRAG organizes retrieval around graph structures for query-focused summarization (Edge et al., 2024). HippoRAG uses neurobiologically inspired graph retrieval for long-term memory (Gutiérrez et al., 2024). Zep represents agent memory with a temporal knowledge graph (Rasmussen et al., 2025). SynapticRAG models temporal memory retrieval with synaptic mechanisms (Hou et al., 2025).

Episodic and event-centric memory. REMem

uses hybrid episodic memory graphs for reasoning over long-form context (Shu et al., 2026). ES-Mem builds long-term dialogue memory with event segmentation (Zou et al., 2026). AnchorMem stores anchored facts with associative contexts for LLM memory (Shen et al., 2026). EM-LLM segments long contexts into retrievable episodic events (Fountas et al., 2025).

Efficient memory and forgetting. LongMem augments language models with long-term memory for long-context use (Wang et al., 2023). FraCom uses a fragment-then-compose framework for long-term conversation memory (Ke et al., 2025). R³Mem studies reversible compression for memory retention and retrieval (Wang et al., 2025).

3 Method

ECO-Mem is an event-centric memory framework for long-term conversational agents. To reduce event fragmentation, it constructs source-linked event episodes as the basic units for storage, retrieval, and pruning. To support dynamic state and correction tracking, it reorganizes episodes with typed event relations that encode temporal flow, state updates, corrections, and continuations. To reduce over-exposure without making pruning irreversible, it separates evidence preservation from the retrieval-visible window. At query time, it retrieves seed events, expands them through relation-guided graph traversal, and reactivates hidden evidence only when source confirmation supports relevance. Figure 1 summarizes the framework.

The following sections instantiate these choices as memory construction, memory maintenance, and query-time reuse.

3.1 Memory Construction

Memory construction transforms a dialogue stream into source-linked event episodes. It first defines the event episode as the basic memory unit and then applies ECO segmentation to decide when continuous turns should be sealed into an event.

Source-linked event episodes. Let $D = \{x_1, \dots, x_T\}$ be a dialogue stream. ECO-Mem represents long-term memory as source-linked event episodes. Each memory unit is an event episode:

$$e_i = (U_i, G_i, B_i, A_i, z_i), \quad (1)$$

where U_i is the source dialogue span, G_i is a local semantic subgraph, B_i is a set of support blocks with source-turn identifiers, A_i is an archive pointer to the raw dialogue evidence, and z_i is the visibility state. A memory bank is a graph $M = (E, \mathcal{R}, Z)$ over event episodes E , typed cross-event relations $\mathcal{R}$, and visibility states Z .

The event episode is the unit for storage, organization, pruning, and retrieval. It keeps the answer-bearing content tied to source dialogue spans instead of turning memory into detached facts or mutable summaries. The visibility state z_i is deliberately part of the memory object: preserving an event and making it retrieval-visible are separate decisions. This allows ECO-Mem to retain recoverable evidence while making only the subset justified by the current query and budget visible for retrieval.

ECO segmentation for event construction. ECO segmentation converts a continuous dialogue into coherent event episodes. It treats an event boundary as a change in the ongoing conversational situation: a topic may shift, a user or world state may be updated, a prior claim may be corrected, or a previously continuous activity may end. The overall procedure is summarized in Algorithm 1; boundary features and thresholds are detailed in Appendix G.1. For each incoming turn, ECO-Mem first reads a lightweight TurnSketch, then compares it with the pending event candidate, and finally decides whether to append the turn or seal the candidate as a completed event. The boundary evidence combines semantic shift, state or correction change, temporal discontinuity, compactness, and token pressure:

$$s_t = g(\Delta_{\text{sem}}, \Delta_{\text{state}}, \Delta_{\text{corr}}, \Delta_{\text{time}}, \Delta_{\text{mdl}}, p_{\text{cp}}, \rho_{\text{tok}}), \quad (2)$$

where Δ_{mdl} denotes the compactness signal, p_{cp} is the changepoint probability, and ρ_{tok} is token pressure. In Figure 1, the boundary detector denotes this scoring-and-thresholding step that decides whether to append the incoming turn or seal the current event candidate. When the boundary score passes the adaptive threshold, the current candidate is sealed as a source-linked event episode for later retrieval. After sealing, ECO-Mem updates the event graph and visibility state before starting the next candidate. If a later event updates an earlier state, corrects a prior claim, or continues an activity that was split across boundaries, this change is recorded through the typed

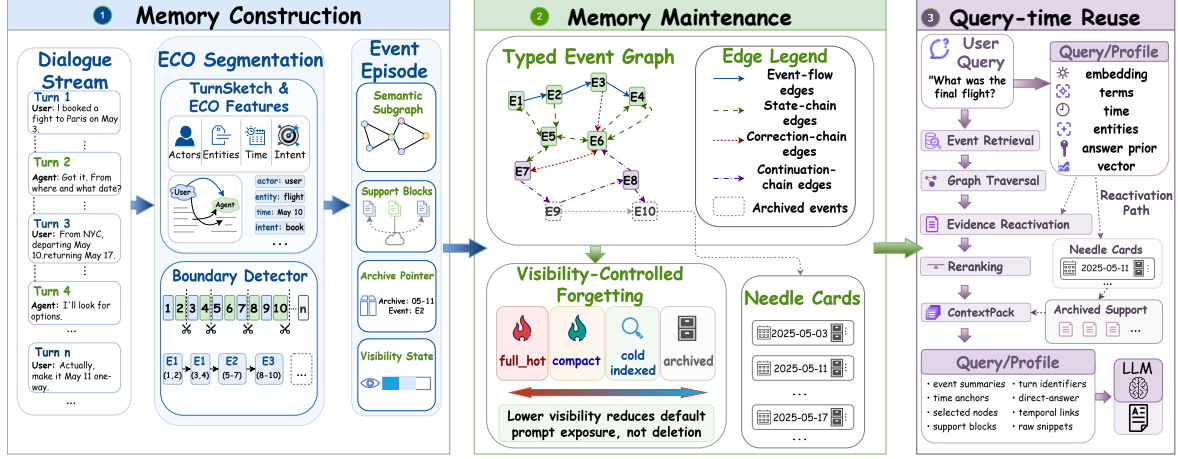


Figure 1: Overview of ECO-Mem. Memory construction converts a dialogue stream into source-linked event episodes with ECO segmentation. Memory maintenance organizes episodes in a typed event graph, controls event visibility through visibility-controlled forgetting, and keeps needle cards for archived events. Query-time reuse builds a query profile, retrieves and traverses events, reactivates archived evidence, reranks support, and packs selected evidence into a compact ContextPack.

Algorithm 1 ECO event segmentation and memory construction

Require: Dialogue turns $x_1, \dots, x_T$

- 1: Initialize pending event candidate c
- 2: **for** x_t in chronological order **do**
- 3: build TurnSketch and ECO features for x_t
- 4: **if** ECOSEGMENT(c, x_t) **then**
- 5: ground and validate the sealed event candidate c
- 6: persist its local evidence subgraph and support blocks
- 7: update cross-event relations and visibility states
- 8: reset c
- 9: **end if**
- 10: append x_t to c
- 11: **end for**
- 12: flush any remaining candidate

event relations described below. In this way, segmentation creates compact event units while the graph preserves event coherence across subsequent changes. The detailed MDL objective, changepoint model, and thresholds are reported in Appendix G, with boundary detection in Appendix G.1 and the event-grounding prompt in Appendix H.

3.2 Memory Maintenance

Memory maintenance reorganizes constructed events after they enter persistent memory. It preserves cross-event coherence through typed relations and controls which preserved events remain in the retrieval-visible window.

Typed event graph reorganization. ECO-Mem organizes grounded episodes with typed event relations. These relations record how later events

develop, update, revise, or continue earlier ones, so that memory remains coherent after the dialogue changes. Four relation families are used. Event-flow edges preserve temporal development between adjacent events. State-chain edges track updates to the same user, entity, attribute, or world state. Correction-chain edges attach denials, revisions, and conflict-resolving statements to the evidence they modify. Continuation-chain edges recover continuity when a long activity is segmented into multiple episodes. These relations allow later retrieval to follow the structure of the conversation rather than relying only on isolated semantic similarity.

Visibility-controlled forgetting. ECO-Mem models forgetting as visibility control over preserved event episodes. Each event receives a visibility state: full, compact, cold, or archived. Lower visibility removes events from the retrieval-visible window but does not erase the event’s source evidence, archive pointer, or graph relations.

Visibility is determined by recency, utility, redundancy, and chain protection. Events that support temporal evolution, state updates, or corrections are protected from aggressive downgrade, while redundant or low-utility events are moved out of the retrieval-visible window. For archived events, ECO-Mem keeps compact *needle cards* as recoverable reactivation handles. A needle card indicates where the system should look when a later query may need hidden evidence. The complete

visibility-control state space, downgrade objective, capacity constraints, and protection rules are provided in Appendix C.

3.3 Query-time Reuse

Query-time reuse converts the maintained memory into a compact evidence package for answer generation. It consists of query profiling and event retrieval, relation-guided graph traversal, visibility-controlled reactivation with source confirmation, and evidence packing.

Query profiling and event retrieval. ECO-Mem first builds a QueryProfile that captures semantic, lexical, temporal, entity, attribute, and answer-type signals. The profile does not choose a single retrieval route; it provides soft priors for event scoring, relation traversal, and evidence packing.

Seed events are retrieved with semantic, lexical, and profile-aware evidence:

$$\text{score}_0(q, e) = \alpha \text{sem}(q, e) + \beta \text{lex}(q, e) + \gamma \text{bias}(q, e), \quad (3)$$

where the bias term captures conversation match, recency, focus entities, attributes, answer-type needs, and temporal signals. Semantic nodes and support blocks can promote their owning event through embedding similarity, target-term match, focus-entity or attribute match, answer-type compatibility, and direct-answer likelihood. Appendix G.2 gives the default scoring weights and runtime diagnostics.

Relation-guided graph traversal. The seed events are then expanded through relation-guided graph traversal. Temporal questions prioritize event-flow links, state questions prioritize state-chain links, correction questions prioritize correction-chain links, and multi-hop or continuity questions prioritize continuation-chain links. These preferences guide traversal without excluding other relevant event paths.

Source-confirmed step of visibility-controlled reactivation. Visibility-controlled reactivation is triggered when a needle card for hidden evidence matches the query. ECO-Mem returns to the archived source support and reactivates hidden evidence only when the source block also confirms relevance through lexical, target-term, or embedding evidence. This source-confirmed step keeps visibility control from becoming irreversible pruning.

Evidence packing. Finally, the expanded episodes are reranked and packed into a

ContextPack. The answer model receives a compact, source-linked evidence package. The answer-generation prompt is provided in Appendix H.

4 Experiments

Datasets. We use the publicly released LoCoMo benchmark for very long-term conversational memory (Maharana et al., 2024). LoCoMo contains multi-session dialogues with long-range dependencies, temporal updates, and questions that require recalling evidence from earlier interactions. This setting directly tests whether a memory system can keep conversation state recoverable without sending the full history to the answer model.

We further evaluate on RealTalk, an open-domain multi-session conversation setting with more natural interaction histories (Lee et al., 2025). This dataset contains 728 questions across three categories: multi-hop (301), temporal (319), and commonsense (108). It tests whether the same event-graph construction and visibility-controlled retrieval remain effective in a less structured open-domain setting. Our RealTalk evaluation uses GPT-4.1-mini-2025-04-14 as the answer model (OpenAI, 2025). Together, LoCoMo and RealTalk evaluate both structured long-term conversational QA and more natural open-domain interaction histories. Table 4 in Appendix A reports the question-type statistics for both evaluation datasets. Appendix D.1 reports the full RealTalk table and analysis as part of the additional results in Appendix D.

Baselines. We compare against strong and representative memory and retrieval strategies that cover the main design families for long-term agent memory. Full Context is a strong upper-cost reference that exposes the complete dialogue history when possible, while NaiveRAG represents standard dense retrieval over text chunks. HippoRAG is a representative graph-RAG baseline for associative retrieval (Gutiérrez et al., 2024). LightMem is a strong lightweight memory baseline that filters, groups, and consolidates memory to reduce online cost (Fang et al., 2026). Zep/Graphiti is a practical temporal knowledge-graph memory baseline with mature engineering support (Rasmussen et al., 2025). REMem builds a hybrid episodic memory graph for reasoning over long-form context (Shu et al., 2026). We report ECO-Mem in two settings: Ours uses all retrieval-visible event

														QA Token	
Model	Method	Overall			Single Hop		Multi Hop		Temporal		Open Domain		Avg. Rank ↓		Use ↓
		F1 ↑	B1 ↑	Jac. ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1	B1	(M)
Qwen3-14B	Full Context	4.78	3.95	4.21	17.46	16.96	1.40	0.73	1.28	0.65	2.26	1.21	8.00	7.75	60.08
	NaiveRAG	7.69	5.88	6.69	<u>28.60</u>	<u>25.23</u>	2.46	1.30	1.81	0.92	3.52	1.82	5.75	5.75	10.94
	HippoRAG	35.25	29.79	26.47	24.89	21.65	26.42	23.79	36.00	33.40	39.44	31.82	<u>3.00</u>	3.75	6.75
	REMem	28.82	26.36	22.76	19.48	16.98	17.34	16.66	31.44	33.76	32.27	27.78	5.50	5.00	3.34
	LightMem	36.20	29.79	28.08	31.62	25.91	22.18	17.43	37.57	30.80	38.81	32.12	<u>3.00</u>	3.25	5.53
	Zep/Graphiti	25.47	20.17	19.71	19.46	15.16	9.85	7.13	7.74	5.43	36.04	28.96	6.00	6.25	<u>4.16</u>
	Ours	42.15	43.66	35.90	25.47	24.73	<u>25.30</u>	<u>26.44</u>	52.58	60.51	45.68	45.54	1.75	1.75	18.25
	Ours(Pruned)	<u>37.86</u>	<u>39.49</u>	<u>31.85</u>	20.37	19.44	23.59	26.79	<u>49.53</u>	<u>56.80</u>	<u>40.89</u>	<u>41.06</u>	<u>3.00</u>	<u>2.50</u>	3.34
Llama-3.1-8B-Instruct	HippoRAG	<u>38.27</u>	34.36	<u>29.88</u>	<u>28.59</u>	<u>26.50</u>	25.38	23.94	32.77	30.05	45.09	<u>39.84</u>	<u>2.50</u>	<u>2.75</u>	5.64
	REMem	29.08	25.89	22.09	20.78	18.41	21.25	19.18	33.05	32.02	31.24	26.83	4.25	4.25	2.57
	LightMem	<u>26.56</u>	<u>25.66</u>	<u>20.56</u>	29.02	28.80	<u>16.44</u>	<u>17.88</u>	<u>13.13</u>	<u>15.49</u>	<u>32.01</u>	<u>29.37</u>	<u>3.75</u>	<u>3.75</u>	<u>4.73</u>
	Ours	35.33	<u>37.67</u>	28.83	22.23	23.59	27.35	<u>30.08</u>	<u>43.08</u>	<u>51.31</u>	37.68	38.06	<u>2.50</u>	<u>2.75</u>	14.71
	Ours(Pruned)	38.32	40.42	31.83	23.54	24.45	<u>25.97</u>	33.80	47.18	54.79	<u>41.31</u>	41.04	2.00	1.50	<u>3.72</u>

Table 1: Main results on LoCoMo by question type. Overall F1, BLEU-1(B1), and Jaccard(Jac.) are percentages; category columns report F1 and B1 for each question type. Higher F1, B1, and Jac. are better. Avg. Rank is computed over the four question types within each backbone group, separately for F1 and B1; lower is better. QA Token Use reports online answer-generation tokens in millions; lower is better. The best result in each backbone group is bolded and the second best is underlined.

episodes, while Ours(Pruned) enables visibility maintenance and compact support projection. Appendix B.1 gives detailed baseline, runtime, ECO-Mem, and ablation settings within the experimental details in Appendix B.

Backbones and evaluation metrics. We run two instruction-tuned backbones: Qwen3-14B (Yang et al., 2025) and Llama-3.1-8B-Instruct (AI@Meta, 2024) for LoCoMo, and GPT-4.1-mini-2025-04-14 (OpenAI, 2025) for RealTalk. To avoid evaluator-model bias when comparing methods on LoCoMo, we use automatic metrics rather than LLM-as-a-judge scores as the main evaluation criteria. Following prior LoCoMo evaluations, we report overall F1, BLEU-1 (B1), and Jaccard, together with category-level F1 and BLEU-1 for single-hop, multi-hop, temporal, and open-domain questions. These metrics measure answer content overlap with the reference answers, with F1 capturing the precision-recall balance; higher F1, BLEU-1, and Jaccard are better. We also report average rank across the four question types within each backbone group, where lower is better. Because long-term agent memory is repeatedly queried after construction, the build cost is paid once, while the cost of reusing past information accumulates through QA-generation tokens. We therefore report online QA token use as the efficiency metric, where lower is better. Table 1 sum-

marizes the LoCoMo results, and Appendix D.1 reports the RealTalk results.

Ablation study. To isolate the contribution of the main ECO-Mem components, we compare the Qwen3-14B Ours(Pruned) setting against ablations that target event segmentation, visibility control, and typed graph traversal, including the source-confirmation step used when hidden evidence is reactivated. For event segmentation, Fixed-8 replaces ECO event boundaries with fixed eight-turn chunks; eight turns is closest to the average number of turns in ECO-segmented events, making the comparison mainly test boundary quality rather than segment length. For visibility control, Recency-token keeps events from newest to oldest until reaching the same 650-token visible-context budget, while Recent-35 keeps only the latest 35 events in each conversation visible. For query-time reuse, 0-hop disables relation-guided graph traversal after seed event retrieval, untyped 2-hop keeps expansion but removes relation-type signals, w/o needle reactivation disables compact hidden-evidence entry points, and needle w/o source check removes source-support confirmation after a needle-card hit. Appendix B.1 describes the ablation configurations in detail.

Variant	F1 Score	F1 Drop	B1 Score	B1 Drop	Jac. Score	Jac. Drop
<i>w/o ECO-S (ECO-Segmentation)</i>						
Fixed-8 segmentation	28.07	25.9%	26.81	32.1%	22.44	29.5%
<i>w/o VSF (Visibility-Controlled Forgetting)</i>						
Recency-token visibility	31.30	17.3%	32.56	17.6%	25.97	18.5%
Recent-35 visibility	32.78	13.4%	33.94	14.1%	27.46	13.8%
<i>w/o GT (Graph Traversal)</i>						
0-hop	34.05	10.1%	36.59	7.3%	31.80	0.2%
<i>w/o TER (Typed Event Relations)</i>						
untyped 2-hop	35.98	5.0%	36.90	6.6%	30.78	3.4%
<i>Visibility-Controlled Reactivation</i>						
w/o needle reactivation	31.56	16.6%	33.10	16.2%	26.21	17.7%
needle w/o source check	37.17	1.8%	38.31	3.0%	31.40	1.4%
ECO-Mem	37.86	–	39.49	–	31.85	–

Table 2: Qwen3-14B ablations on LoCoMo; Drop is relative to ECO-Mem.

5 Analysis

Our analysis is organized around four research questions. RQ1 asks whether ECO-Mem improves long-term conversational ability compared with existing memory and retrieval frameworks. RQ2 asks whether event-centric memory organization and visibility-controlled pruning reduce cost and noise while preserving answer quality. RQ3 asks how much each core component contributes, including ECO event segmentation, visibility-controlled forgetting, relation-guided graph traversal, and the source-confirmed step of visibility-controlled reactivation. RQ4 asks whether the event retriever and graph expansion settings are robust to reasonable parameter changes.

Main Results (RQ1). Table 1 shows that organizing memory around events is effective for long-term conversational QA. With Qwen3-14B, the unpruned ECO-Mem obtains the best overall F1, BLEU-1, and Jaccard, and the pruned variant remains second best overall while using far fewer QA tokens. The strongest gains appear on temporal and open-domain questions, where the answer model must recover evidence distributed across turns or reason over changing states. The pruned Qwen3-14B variant is weaker on single-hop questions, indicating that aggressive visibility-controlled forgetting can downgrade events that lack strong direct-answer support. This answers RQ1 positively: compared with full-context prompting, chunk retrieval, graph retrieval, and lightweight memory baselines, ECO-Mem improves long-term dialogue QA by keeping event-level evidence coherent and recoverable.

The Llama-3.1-8B-Instruct results make the

pruning effect clearer under a smaller local backbone. Ours(Pruned) achieves the best overall F1, BLEU-1, and Jaccard while using much less QA budget than the unpruned variant. Its gains indicate that the model benefits from fewer but more relevant event supports. This indicates that visibility-controlled pruning reduces not only cost but also retrieval noise, an effect that becomes more important when the answer model has weaker reasoning capacity. Appendix D.3 provides representative examples from the four LoCoMo question types, and Appendix D.1 reports the full RealTalk table.

As shown in Table 5, on RealTalk with **GPT-4.1-mini-2025-04-14** as the answer backbone, ECO-Mem obtains the best overall F1 and BLEU-1 (26.8 and 28.7) and the best temporal F1 and BLEU-1 (34.7 and 39.7), demonstrating transfer to natural multi-session histories.

Efficiency Analysis (RQ2). Figure 2 shows that ECO-Mem reaches the best quality-efficiency balance among the compared frameworks on both backbones. In the Qwen3-14B setting, Ours gives the strongest quality, while Ours(Pruned) moves to a more efficient region by reducing QA token use from 18.25M to 3.34M, an about 82% reduction, while retaining nearly 90% of the unpruned overall F1. This places the ECO-Mem variants above the other methods in the Qwen3-14B quality-efficiency comparison. In the Llama-3.1-8B-Instruct setting, Ours(Pruned) is even more favorable: it improves overall quality over the unpruned variant and uses a much smaller QA-time budget, showing that visibility control can reduce noise as well as cost. REMem and Ours(Pruned) both show favorable quality-efficiency balance in this setting, while Ours(Pruned) trades a higher but still efficient QA budget for the strongest overall quality.

Tables 10 and 11 in Appendix E show that the quality–efficiency trade-off persists across GPT-4o and Qwen3 model scales. Including one-time construction cost, Ours(Pruned) reaches the lowest amortized cost in both backbone groups from four reuse rounds onward; the complete cost analysis is provided in Appendix E.3. These results answer RQ2: event-centric organization with visibility-controlled pruning successfully reduces cost and noise while preserving a strong quality-efficiency trade-off.

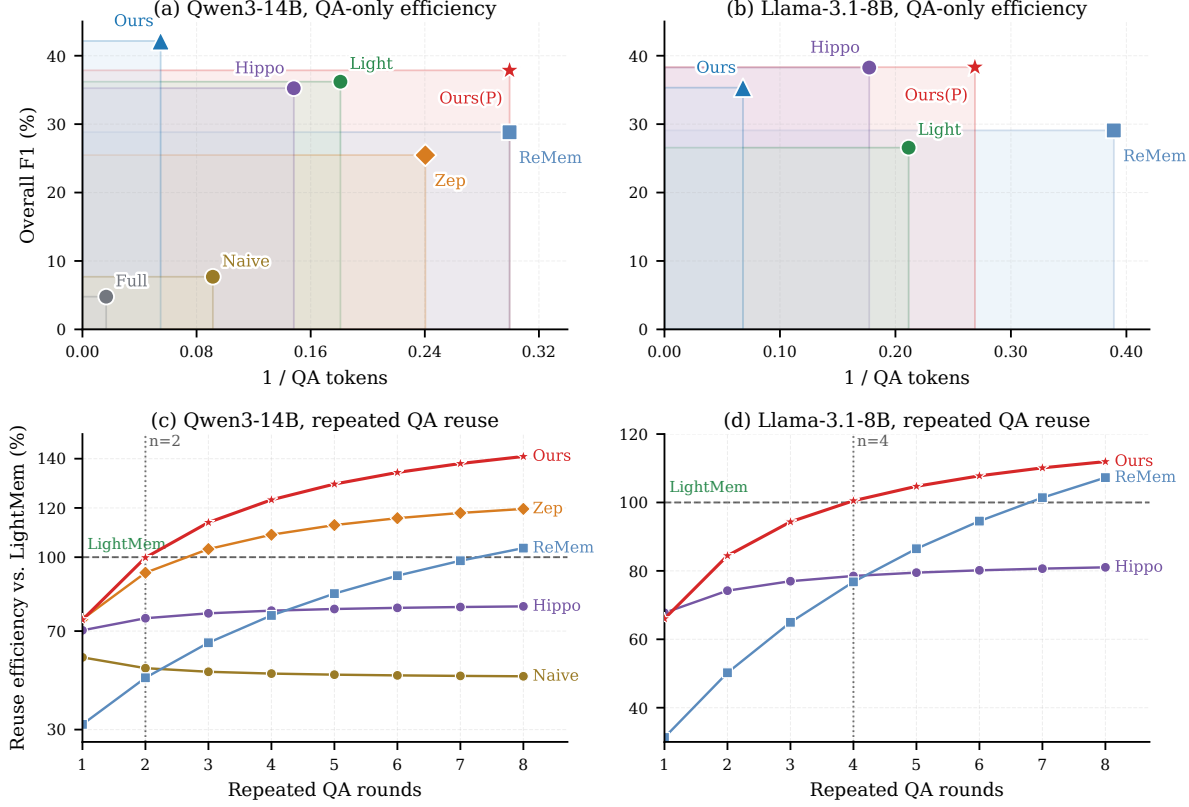


Figure 2: QA-only quality-efficiency and repeated-reuse token amortization on LoCoMo. Panels (a) and (b) compare Qwen3-14B and Llama-3.1-8B-Instruct using QA-generation tokens only. Panels (c) and (d) compare repeated-reuse efficiency against LightMem when the same constructed memory is reused for multiple QA rounds. Higher is better in all panels.

Ablation Study (RQ3). Table 2 and Figure 3 ablate the key components of ECO-Mem. Fixed-8 produces the largest overall drop, supporting the event fragmentation claim: the loss is not only a matter of chunk size, but of whether boundaries preserve coherent conversational events. The two time-based visibility baselines also fall below full ECO-Mem, supporting our pruning strategy: visibility should be selected by event utility, state changes, corrections, and recoverability rather than by time alone.

The 0-hop graph traversal ablation causes a smaller overall drop, but the degradation is concentrated on multi-hop and open-domain questions, where relation-based expansion is more important. Removing the relation-type signal while keeping the graph and 2-hop expansion intact degrades all three metrics, showing that typed event relations are operational signals used by relation-guided traversal, not decorative labels. The needle reactivation ablation causes larger overall and open-domain drops, suggesting that hidden evidence required by cross-event questions is less

Method	Overall F1	BLEU-1
ES-Mem (paper)	30.46	25.48
ECO-Mem (noprun)	26.31	32.53
ECO-Mem (pruned)	33.15	39.85

Table 3: Same-backbone comparison under Llama3.2-3B. ES-Mem results are from its original paper; ECO-Mem results use the same answer backbone.

likely to be recovered without compact reactivation entry points. Together, these results answer RQ3: event segmentation, visibility control, typed graph traversal, and its source-confirmed step of visibility-controlled reactivation all contribute to long-term memory interaction. Under the same Llama3.2-3B answer backbone reported by ES-Mem’s original paper, ECO-Mem (pruned) outperforms ES-Mem by 2.69 overall F1 and 14.37 BLEU-1 (Zou et al., 2026); Table 3 gives the comparison. Appendix D.2 provides the category-level F1 breakdown used in this analysis.

Parameter Sensitivity (RQ4). We design multiple parameter experiments to evaluate the robust-

ness of ECO-Mem. Appendix F.1 varies graph expansion depth, while Table 14 in Appendix F.2 varies the seed retrieval weights. The results remain close across these settings. Different weights shift small category-level trade-offs, but the overall performance remains stable. This suggests that the system’s gains are not driven by a single fragile parameter choice, though broader sweeps would be needed to fully characterize their limits.

6 Conclusion

We presented ECO-Mem, an event-centric organization framework for visibility-controlled long-term memory. We frame long-term agent memory around three linked challenges: full memory reuse can increase retrieval noise and repeated QA-time cost, long contexts can miss key information, and structured storage can split dialogue evidence into isolated units. ECO-Mem responds to these challenges by constructing source-linked event episodes, reorganizing them into a typed event graph, and applying visibility-controlled forgetting that hides redundant or low-utility events without erasing their source evidence or graph relations. At query time, relation-guided graph traversal followed by visibility-controlled reactivation with source confirmation recovers compact support for generation. Comparisons with existing methods on LoCoMo and RealTalk, together with multiple ablation studies, demonstrate the effectiveness and efficiency of ECO-Mem for long-term agent memory. These findings suggest that recoverable event evidence and visibility control are useful directions for building more efficient long-term agent memory.

Limitations

Although ECO-Mem improves long-term conversational memory, several limitations remain. First, the framework relies on online event segmentation and LLM-based grounding. Gradual topic drift, interleaved conversations, or ambiguous corrections may cause over- or under-segmentation, while grounding may omit fine-grained facts or introduce noisy semantic nodes. These errors can propagate to graph construction, visibility maintenance, and retrieval.

Second, ECO-Mem trades upfront memory construction cost for lower query-time exposure. This design is most suitable when a persistent memory is reused across many future queries. For

one-shot use, very short histories, or settings with limited construction budget, lighter retrieval-only memory systems may be more economical. In addition, the current implementation uses fixed budgets and hand-set scoring weights, which may require adaptation for domains with different dialogue lengths or evidence density.

Third, visibility-controlled forgetting is non-destructive but not infallible. Archived evidence remains recoverable only when query profiling, relation-guided graph traversal, and evidence reactivation identify the right entry points. Questions that depend on underspecified, implicit, or weakly lexicalized evidence may still fail to reactivate relevant hidden events.

Ethical Considerations

We use public benchmarks and collect no new human-subject data or personally identifiable information. Datasets and models are used for research under their respective licenses. Because long-term memory systems may retain sensitive information across sessions, deployments should obtain consent and provide retention limits, deletion mechanisms, access control, auditability, and data minimization. Misuse risks should also be assessed before deployment in high-stakes settings such as mental health, finance, and workplace monitoring.

Use of AI Assistants

We used general-purpose AI assistants for language polishing and limited productivity support under author supervision. They were not used to generate research ideas, design experiments, make scientific decisions, or produce the paper’s technical claims, results, or conclusions. All technical content, experimental results, and conclusions are the authors’ own, and the authors take full responsibility for the correctness of the paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No. 2025YFB3003700) and the National Natural Science Foundation of China (Grant No. 62476025).

References

- AI@Meta. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Alan Baddeley. 1992. [Working memory](#). *Science*, 255(5044):556–559.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. 2023. [Extending context window of large language models via positional interpolation](#). *Preprint*, arXiv:2306.15595.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. [Mem0: Building production-ready AI agents with scalable long-term memory](#). In *ECAI 2025*. IOS Press.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. [From local to global: A graph RAG approach to query-focused summarization](#). *Preprint*, arXiv:2404.16130.
- Jizhan Fang, Xinle Deng, Haoming Xu, Ziyang Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. 2026. [Lightmem: Lightweight and efficient memory-augmented generation](#). In *The Fourteenth International Conference on Learning Representations*.
- Zafeirios Fountas, Martin A. Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. 2025. [Human-inspired episodic memory for infinite context LLMs](#). In *The Thirteenth International Conference on Learning Representations*.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. [Hipporag: Neurobiologically inspired long-term memory for large language models](#). In *Advances in Neural Information Processing Systems 37*, pages 59532–59569. Neural Information Processing Systems Foundation, Inc.
- Yuki Hou, Haruki Tamoto, Qinghua Zhao, and Homei Miyashita. 2025. [SynapticRAG: Enhancing temporal memory retrieval in large language models through synaptic mechanisms](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20422–20436, Vienna, Austria. Association for Computational Linguistics.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. [Memory in the age of AI agents](#). *Preprint*, arXiv:2512.13564.
- Cai Ke, Yiming Du, Bin Liang, Yifan Xiang, Lin Gui, Zhongyang Li, Baojun Wang, Yue Yu, Hui Wang, Kam-Fai Wong, and Ruifeng Xu. 2025. [Flexibly utilize memory for long-term conversation via a fragment-then-compose framework](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21119–21136, Suzhou, China. Association for Computational Linguistics.
- Dong-Ho Lee, Adyasha Maharana, Jay Pujara, Xiang Ren, and Francesco Barbieri. 2025. [REALTALK: A 21-day real-world dataset for long-term conversation](#). *Preprint*, arXiv:2502.13270.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. [Evaluating very long-term conversational memory of LLM agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.
- Simon Nørby. 2015. [Why forget? on the adaptive value of memory loss](#). *Perspectives on Psychological Science*, 10(5):551–578.
- OpenAI. 2025. [GPT-4.1 in the API](#). OpenAI. Accessed 2026-05-25.
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2023. [MemGPT: Towards LLMs as operating systems](#). *Preprint*, arXiv:2310.08560.
- Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA. Association for Computing Machinery.

- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. [Zep: A temporal knowledge graph architecture for agent memory](#). *Preprint*, arXiv:2501.13956.
- Blake A. Richards and Paul W. Frankland. 2017. [The persistence and transience of memory](#). *Neuron*, 94(6):1071–1084.
- Zhanyu Shen, Sijie Cheng, Zhicheng Guo, Weiqin Wang, Yile Wang, and Hui Huang. 2026. [Anchormem: Anchored facts with associative contexts for building memory in large language models](#). *Preprint*, arXiv:2604.17377.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems 36*, pages 8634–8652. Neural Information Processing Systems Foundation, Inc.
- Yiheng Shu, Saisri Padmaja Jonnalagedda, Xiang Gao, Bernal Jiménez Gutiérrez, Weijian Qi, Kamalika Das, Huan Sun, and Yu Su. 2026. [REMem: Reasoning with episodic memory in language agent](#). In *The Fourteenth International Conference on Learning Representations*.
- Jianlin Su, Ahmed Murtadha, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2024. [Roformer: Enhanced transformer with rotary position embedding](#). *Neurocomputing*, 568:127063.
- Endel Tulving and Zena Pearlstone. 1966. [Availability versus accessibility of information in memory for words](#). *Journal of Verbal Learning and Verbal Behavior*, 5(4):381–391.
- Endel Tulving and Donald M. Thomson. 1973. [Encoding specificity and retrieval processes in episodic memory](#). *Psychological Review*, 80(5):352–373.
- Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023. [Augmenting language models with long-term memory](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 74530–74543. Curran Associates, Inc.
- Xiaoqiang Wang, Suyuchen Wang, Yun Zhu, and Bang Liu. 2025. [R³Mem: Bridging memory retention and retrieval via reversible compression](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4541–4557. Association for Computational Linguistics.
- Yu Wang and Xi Chen. 2025. [MIRIX: Multi-agent memory system for LLM-based agents](#). *Preprint*, arXiv:2507.07957.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. [A-MEM: Agentic memory for LLM agents](#). In *Advances in Neural Information Processing Systems 38*, pages 20004–20031. Neural Information Processing Systems Foundation, Inc.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *International Conference on Learning Representations*.
- Jeffrey M. Zacks, Nicole K. Speer, Khen M. Swallow, Todd S. Braver, and Jeremy R. Reynolds. 2007. [Event perception: A mind-brain perspective](#). *Psychological Bulletin*, 133(2):273–293.
- Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Jirong Wen. 2025. [A survey on the memory mechanism of large language model-based agents](#). *ACM Transactions on Information Systems*, 43(6):1–47.
- Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. [MemoryBank: Enhancing large language models with long-term memory](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, pages 19724–19731. AAAI Press.
- Huhai Zou, Tianhao Sun, Chuanjiang He, Yu Tian, Zhenyang Li, Li Jin, Nanyu Liu, Jiang Zhong, and Kaiwen Wei. 2026. [ES-Mem: Event segmentation-based memory for long-term dialogue agents](#). *Preprint*, arXiv:2601.07582.

A Dataset Question Statistics

Dataset	Question type	Count
LoCoMo	Single Hop	282
	Multi Hop	96
	Temporal	321
	Open Domain	841
	Total	1540
RealTalk	Category 1: Multi-hop	301
	Category 2: Temporal	319
	Category 3: Commonsense	108
	Total	728

Table 4: Question-type statistics for the evaluated datasets. LoCoMo counts cover the four categories used in the main evaluation; the adversarial category is not evaluated. RealTalk contains 728 questions; Categories 1, 2, and 3 contain 301, 319, and 108 questions, respectively. These counts follow the evaluated question set used in Appendix D.1.

B Experimental Details

B.1 Baseline and Ablation Settings

Full Context. *Method.* Full Context is a direct long-context baseline. It does not build a memory structure, execute retrieval or pruning, or vectorize the dialogue history. Instead, it feeds the complete dialogue history of each LoCoMo sample directly to the answer model, testing whether simply enlarging the context window and reading the whole history can replace long-term conversational memory retrieval. *Setting.* We use Qwen3-14B (Yang et al., 2025) for the Qwen group and Llama-3.1-8B-Instruct (AI@Meta, 2024) for the Llama group. Because the tokenized dialogue history together with the chat template can exceed the default serving window, we enable RoPE-based context extension and set the position window to 65,536 tokens, following rotary position embeddings and position interpolation for long-context extrapolation (Su et al., 2024; Chen et al., 2023).

NaiveRAG. *Method.* NaiveRAG is a standard retrieval-augmented generation baseline (Lewis et al., 2020). It does not construct an event graph, perform event boundary detection, or apply visibility pruning. It directly splits the raw dialogue history into fixed-length text chunks, retrieves chunks by vector similarity to the query, and provides the retrieved text to the answer model. This baseline tests how far plain text-chunk retrieval can go in long-term conversational QA. *Setting.* Unless otherwise stated, chunks are embedded with BGE-

M3 and retrieved with semantic similarity. Retrieved chunks are inserted into the same answer-generation template as other local baselines, using deterministic decoding.

HippoRAG. *Method.* HippoRAG (Gutiérrez et al., 2024) is a representative graph-RAG baseline for associative retrieval. It extracts structured signals from text, links them through graph-style associations, and uses graph traversal to complement dense retrieval when answering questions. *Setting.* We use its graph-augmented retrieval output as the evidence context for the same answer backbones used in the corresponding LoCoMo groups.

LightMem. *Method.* LightMem (Fang et al., 2026) is a strong lightweight memory baseline based on a fragment-then-compose design. It filters perceived information online, groups short-term memory fragments, and consolidates them into compact long-term memory to reduce online cost. *Setting.* We run LightMem with the same local answer backbones as the corresponding LoCoMo groups and report its generated-answer metrics and QA token use under the shared evaluation protocol.

Zep/Graphiti. *Method.* Zep/Graphiti (Rasmussen et al., 2025) is a practical temporal knowledge-graph memory baseline. It extracts conversation facts, organizes them with temporal and relational metadata, and retrieves graph facts for later queries. *Setting.* We evaluate its retrieved graph facts as the memory evidence for answer generation under the same LoCoMo question categories.

REMem. *Method.* REMem (Shu et al., 2026) builds an episodic memory graph for language agents, combining episodic records with graph reasoning so that long-form interaction history can be reused in downstream reasoning. *Setting.* We use the REMem-S setting for RealTalk results reported from the original paper. For LoCoMo, we run the public REMem code with the same local answer backbones and evaluate its generated answers with the same scoring protocol as other local baselines.

Common runtime settings. All locally run LoCoMo experiments use the validation split with question categories 1–4. The Qwen group uses Qwen3-14B (Yang et al., 2025), and the Llama group uses Llama-3.1-8B-Instruct (AI@Meta,

2024). The local backbones therefore use 14B and 8B parameters, respectively, while RealTalk uses GPT-4.1-mini-2025-04-14 (OpenAI, 2025) as the answer model. Local inference and memory construction are run on four NVIDIA RTX A6000 GPUs. Unless a baseline requires its own encoder, retrieval embeddings use BGE-M3 with a maximum embedding length of 512. Answer generation uses deterministic decoding with temperature 0 and a maximum of 1024 generated tokens. For ECO-Mem retrieval, the default query-time limits are seed- $k = 12$, graph depth 2, at most 8 context events, at most 16 context blocks, and a 650-word strict evidence budget for pruned variants.

ECO-Mem settings. The unpruned ECO-Mem setting builds source-linked event episodes and typed event relations, with all retrieval-visible event episodes available during retrieval. The Qwen3-14B unpruned row in Table 1 uses the corresponding no-pruning configuration. Ours(Pruned) uses the same event graph but enables visibility-controlled maintenance with a strict query-time evidence budget, needle-card indexing, source-support confirmation, relation-guided graph traversal, and evidence reranking before ContextPack generation. The Qwen3-14B pruned row reports the completed pruned evaluation after all validation questions are processed.

Ablation settings. The Fixed-8 segmentation ablation replaces ECO event segmentation with fixed eight-turn chunks. We choose eight turns because it is closest to the average event length produced by ECO event segmentation, so the ablation keeps segment granularity comparable while removing adaptive event-boundary detection. The w/o GT 0-hop ablation keeps seed event retrieval but disables relation-guided graph traversal before evidence reranking and ContextPack generation. The untyped 2-hop ablation keeps the graph and 2-hop expansion fixed but replaces all relation types with a single generic edge type. The Recency-token visibility ablation removes Visibility-Controlled Forgetting by keeping only the most recent retrieval-visible support blocks under approximately the same 650-token strict evidence budget, with hidden evidence reactivation disabled. The Recent-35 visibility ablation is a stronger recency baseline that marks the most recent 35 events in each conversation as visible and archives earlier events. The w/o nee-

dle reactivation ablation keeps the pruned visibility states fixed but disables needle-card indexing and hidden evidence reactivation, isolating the effect of visibility-controlled reactivation at query time. The needle w/o source check ablation keeps needle-card indexing and reopening enabled, but removes the source-support confirmation required after a direct needle-card hit. This ablation uses the Qwen3-14B Ours(Pruned) setting and holds the remaining retrieval settings fixed.

C Formal Visibility-Control Procedure

State space and capacities. Each event e_i carries a visibility state $z_i \in \{\text{full}, \text{compact}, \text{cold}, \text{archived}\}$. The first three levels have hard capacities K_1 , K_2 , and K_3 , with defaults 5, 10, and 20; events beyond $K_1 + K_2 + K_3$ are archived. Each lower level applies a decreasing exposure multiplier, so a downgraded event contributes fewer tokens to the retrieval-visible window while its archive pointer A_i , needle card, and typed relations remain intact.

Downgrade objective. For a candidate downgrade of e_i , the maintenance procedure computes

$$s(e_i) = \frac{D(e_i) - \lambda_r R(e_i)}{C(e_i)}, \quad (4)$$

where $D(e_i) = L_{\text{sup}} + L_{\text{chain}} + L_{\text{time}} + L_{\text{corr}} + L_{\text{ret}}$ combines support-block coverage loss, chain-relation coverage, temporal reachability, correction-chain targets, and observed retrieval usage. $R(e_i)$ is the fraction of the event’s support atoms covered by neighboring events, using $k = 4$ neighbors by default; $C(e_i)$ is the exposure reduction under the target level; and $\lambda_r = 0.35$. The procedure repeatedly selects the lowest-scoring candidate and downgrades it until every level satisfies its capacity.

Protection and non-destructiveness. An event-flow head, latest state-chain node, correction-chain target, or continuation-chain head cannot descend below compact. An event whose support-block atoms are not covered elsewhere also carries a support floor. Lowering visibility never deletes source evidence: A_i and the needle card remain available, and the source-confirmed step of visibility-controlled reactivation reopens a hidden event only when the source block confirms a needle-card hit.

D Additional Results

D.1 RealTalk Results

The RealTalk results test the same memory design on open-domain multi-session interaction histories. ECO-Mem obtains the best overall F1 and BLEU-1, with the clearest advantage on temporal questions. This pattern is consistent with

the method’s event segmentation and typed temporal/state relations: the system is especially effective when answers depend on locating the right event and preserving its time-sensitive context. The multi-hop row is less dominant, suggesting that some RealTalk questions require broader reasoning beyond the retrieved event episodes.

Method	Overall		Multi-hop		Commonsense		Temporal	
	F1	B1	F1	B1	F1	B1	F1	B1
Full-Context	25.3	18.6	<u>27.8</u>	<u>21.7</u>	19.5	<u>15.1</u>	24.9	16.8
Qwen3-Embed-8B	20.2	14.9	19.6	16.2	14.5	11.8	22.7	14.8
NV-Embed-v2 (7B)	23.8	17.7	24.5	19.4	16.6	13.0	25.6	<u>17.7</u>
Mem0	9.8	7.2	14.0	9.6	12.0	9.0	5.1	4.3
Graphiti	15.1	11.5	19.6	15.5	16.7	13.2	10.5	7.1
HippoRAG 2	21.9	16.2	26.1	20.9	18.2	13.7	19.2	12.6
REMem-S	<u>26.2</u>	<u>19.2</u>	28.6	23.1	18.2	14.7	<u>26.7</u>	17.1
Ours	26.8	28.7	20.9	20.5	<u>18.9</u>	17.9	34.7	39.7

Table 5: Performance (%) on RealTalk. Ours uses **GPT-4.1-mini-2025-04-14** as the answer model. The baseline rows are adapted from Table 11 of Shu et al. (2026); LLM-as-a-judge scores are omitted. B1 denotes BLEU-1.

D.2 Category-Level Ablation Results

Figure 3 breaks the component ablations down by question type. Replacing ECO-Segmentation with fixed-8 segmentation hurts all categories, showing that coherent event boundaries matter even for direct lookup questions. Removing graph traversal with 0-hop seed retrieval is most visible on multi-hop and open-domain questions, where relation-based expansion helps connect dispersed events after the initial retrieval step. Recency-only visibility is most damaging on temporal and open-domain questions, where older but still relevant

evidence must remain recoverable. Disabling the source-confirmed step of visibility-controlled reactivation leaves temporal questions largely intact, but substantially weakens open-domain answers, indicating that reactivating hidden evidence is most useful when a question requires combining several dispersed pieces of evidence. Removing only the source-support check after a needle-card hit remains close to full ECO-Mem, but still underperforms it on open-domain questions, suggesting that source confirmation mainly improves the precision of recovered evidence.

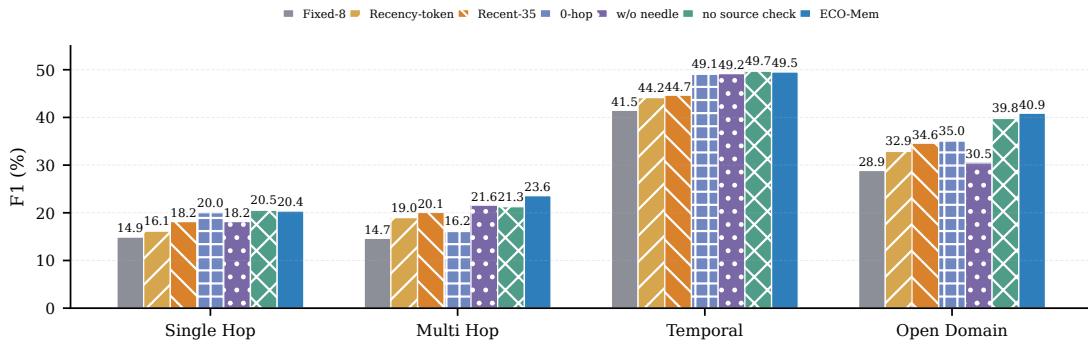


Figure 3: Category-level F1 for the component ablations in Table 2. The grouped bars show how replacing ECO event segmentation with fixed-size chunks, replacing graph traversal with 0-hop seed retrieval, replacing Visibility-Controlled Forgetting with recency heuristics, disabling needle-card reactivation, or removing source-support confirmation affects each question type.

D.3 Qualitative Case Study

Tables 6–9 present four completed examples from the Qwen3-14B Ours(Pruned) run, selected from different LoCoMo question types. The cases show

Question	What career path has Caroline decided to pursue?
Gold	counseling or mental health for transgender people
ECO-Mem	counseling and mental health
Evidence pattern	State-chain retrieval. The ContextPack includes the early education plan (D1:9), direct counseling and mental-health interest (D1:11, D4:11), and the later focus on supporting trans people (D4:13). The answer preserves the stable career state without exposing all related history.

Table 6: Single-hop case from Qwen3-14B Ours(Pruned).

Question	When did Caroline apply to adoption agencies?
Gold	the week of 23 August 2023
ECO-Mem	the week of 23 August 2023
Evidence pattern	Temporal normalization. The direct application event (D13:1) contains “this week”; the temporal hint resolves it to the week of 23 August 2023. The same event chain also retrieves earlier agency research (D2:8) and later interview progress (D19:1), preventing confusion between adjacent adoption milestones.

Table 8: Temporal case from Qwen3-14B Ours(Pruned).

E Additional Answer-Model and Cost Analyses

E.1 Answer-Model Transfer on Conversation 26

Answer model	Mode	F1	B1	Jac.	QA tokens (M)
Qwen3-14B	noprune	42.46	44.06	36.25	1.78
Qwen3-14B	pruned	38.56	41.56	33.30	0.31
GPT-4o	noprune	44.00	42.00	37.70	1.50
GPT-4o	pruned	42.40	42.00	36.50	0.29
GPT-4o	Full-Context	44.80	41.70	35.30	2.43

Table 10: Controlled answer-model results on LoCoMo conversation 26. B1 denotes BLEU-1 and Jac. denotes Jaccard.

Swapping only the answer model from Qwen3-14B to GPT-4o with the same memory graph raises F1 from 42.46 to 44.00 (noprune) and from 38.56 to 42.40 (pruned). GPT-4o with ECO-Mem’s pruned memory reaches 42.40 F1, within 2.40 points of GPT-4o Full-Context (44.80), while us-

how ECO-Mem uses source-linked event episodes for direct state recall, cross-event reasoning, temporal normalization, and open-domain evidence composition.

Question	Would Caroline still want to pursue counseling as a career if she had not received support growing up?
Gold	Likely no
ECO-Mem	It is unlikely that Caroline would still want to pursue counseling without that support
Evidence pattern	Cross-event causal link. Relation-guided graph traversal connects her stated career goal (D4:11, D4:13) with later evidence that received support made her want to help others (D7:7). This lets the model answer a counterfactual question rather than copy a single fact.

Table 7: Multi-hop case from Qwen3-14B Ours(Pruned).

Question	How does Gina stay confident in her business?
Gold	by remembering successes and progress, relying on support, and focusing on why she started
ECO-Mem	by remembering successes and progress, building relationships, staying positive, and using support
Evidence pattern	Evidence composition. The answer combines Gina’s direct confidence statement (D10:8), business relationship and brand advice (D7:6), and her motivation from dance and fashion (D8:6). The result is a compact source-supported synthesis rather than a broad summary.

Table 9: Open-domain case from Qwen3-14B Ours(Pruned). Source turn identifiers in Tables 6–9 come from the Qwen3-14B Ours(Pruned) evaluation; evidence descriptions are shortened from retrieved source-linked support blocks.

ing only 11.9% as many QA tokens (0.29M vs. 2.43M, an 88.1% reduction).

E.2 Controlled Qwen3 Scaling on Full LoCoMo

Answer model	Mode	F1	B1	Jac.	QA tokens (M)
Qwen3-4B	noprune	37.20	40.10	34.20	18.86
Qwen3-4B	pruned	39.00	41.60	36.10	3.80
Qwen3-8B	noprune	40.20	41.70	35.80	18.18
Qwen3-8B	pruned	39.90	42.00	37.40	3.02
Qwen3-14B	noprune	42.15	43.66	35.90	18.25
Qwen3-14B	pruned	37.86	39.49	31.85	3.34

Table 11: Controlled Qwen3-family scaling on full LoCoMo.

Pruning reduces QA tokens by approximately 80% at every scale (about 18M to 3–4M). The F1 changes are +1.80 at 4B, −0.30 at 8B, and −4.29 at 14B.

E.3 Construction Cost and Amortization

For n reuse rounds, the amortized chat-token cost is $C(n) = B + nQ$, where B is one-time memory-construction cost and Q is one QA-round cost. All values in Table 12 are millions of chat tokens and exclude embedding and grading.

Backbone	Method	B	Q	$n = 1$	$n = 2$	$n = 4$	$n = 8$
Qwen3-14B	NaiveRAG	0.00	10.94	10.94	21.88	43.76	87.52
	HippoRAG	2.48	6.75	9.23	15.98	29.48	56.48
	REMem	16.87	3.34	20.21	23.55	30.23	43.59
	Zep/Graphiti	4.51	4.16	8.67	12.83	21.15	37.79
	LightMem	0.96	5.53	6.49	12.02	23.08	45.20
	Ours (P)	5.36	3.34	8.70	12.04	18.72	32.08
Llama-3.1-8B	HippoRAG	2.76	5.64	8.40	14.04	25.32	47.88
	REMem	15.61	2.57	18.18	20.75	25.89	36.17
	LightMem	0.96	4.73	5.69	10.42	19.88	38.80
	Ours (P)	4.90	3.72	8.62	12.34	19.78	34.66

Table 12: Construction and repeated-reuse chat-token cost in millions.

ECO-Mem (P) reaches the lowest amortized cost in both backbone groups from $n = 4$ onward, confirming that visibility-controlled pruning trades a modest upfront build cost for lower cost under persistent reuse.

F Parameter Sensitivity

F.1 Graph-Hop Parameter Sensitivity

Table 13 shows that the default 2-hop expansion achieves the best overall F1 (38.32) and Jaccard (31.83), improving overall F1 by 2.27 points over 0-hop and 2.55 points over 1-hop. The largest gain appears on open-domain questions, where 2-hop

reaches 41.31 F1 compared with 37.41 and 37.17 for 0-hop and 1-hop, respectively. Although 0-hop remains slightly stronger on temporal and multi-hop F1, the aggregate results indicate that limited graph expansion improves evidence composition without requiring deep traversal.

Graph hops	Overall			Single Hop		Multi Hop		Temporal		Open Domain	
	F1 ↑	B1 ↑	Jac. ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑
0-hop	36.05	39.59	30.65	22.31	24.65	26.65	32.40	47.35	55.36	37.41	39.41
1-hop	35.77	39.39	30.29	22.34	24.87	26.41	31.92	46.72	54.61	37.17	39.31
2-hop (default)	38.32	40.42	31.83	23.54	24.45	25.97	33.80	47.18	54.79	41.31	41.04

Table 13: Graph expansion depth sensitivity under Llama-3.1-8B-Instruct Ours(Pruned) on LoCoMo. The default 2-hop setting matches Table 1. B1 denotes BLEU-1 and Jac. denotes Jaccard.

F.2 Seed Retrieval Weight Sensitivity

We further vary the linear weights in the seed event retrieval scorer $\text{score}_0(q, e) = \alpha \text{sem}(q, e) + \beta \text{lex}(q, e) + \gamma \text{bias}(q, e)$. Here α controls dense

semantic similarity, β controls lexical target-term matching, and γ controls profile-aware bias from the query profile. All other retrieval and pruning settings are kept fixed.

Setting	α	β	γ	Overall			Single Hop		Multi Hop		Temporal		Open Domain	
				F1 ↑	B1 ↑	Jac. ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑	F1 ↑	B1 ↑
semantic-only	1.00	0.00	0.00	36.54	40.02	30.62	22.97	25.00	26.13	31.28	46.93	54.70	38.32	40.44
no-bias	0.67	0.33	0.00	36.46	39.86	30.34	22.56	24.67	27.47	32.12	47.16	54.65	38.06	40.20
default	0.60	0.30	0.10	36.59	40.03	30.59	23.09	25.10	28.26	33.42	46.71	54.35	38.21	40.32
lexical-heavy	0.45	0.45	0.10	36.37	39.80	30.30	23.03	25.10	27.51	33.14	47.00	54.89	37.79	39.73
bias-heavy	0.50	0.25	0.25	36.78	40.30	30.75	23.39	25.71	28.41	33.29	47.94	55.75	37.96	40.10

Table 14: Seed retrieval scoring weight sensitivity under Llama-3.1-8B-Instruct Ours(Pruned) on LoCoMo. B1 denotes BLEU-1 and Jac. denotes Jaccard.

G Implementation Details

G.1 Boundary Detection Details

For each candidate event c_{t-1} and incoming turn x_t , the online detector computes a feature vector

$$\delta_t = [d_s, d_l, d_e, d_r, d_\tau, d_c, d_g, \rho_t, \Delta_t, p_t],$$

where the terms measure semantic surprise, local semantic shift, entity, role, temporal, and claim shifts, timestamp-gap outliers, token pressure, MDL-style split gain, and changepoint probability. For a set of turn features A , the compactness cost is

$$\mathcal{C}(A) = |A| [\log(1 + \text{Var}_z(A)) + H(E_A) + 0.3H(R_A) + 0.2H(T_A)],$$

and the split gain is

$$\Delta_t = \mathcal{C}(c_{t-1} \cup x_t) - \mathcal{C}(c_{t-1}) - \mathcal{C}(x_t) - \lambda_b.$$

The detector combines normalized boundary signals with token pressure, compactness gain, and changepoint probability, and flushes a candidate only when the adaptive threshold, minimum length, token, and cooldown constraints are satisfied.

G.2 Query-Time Scoring Details

For seed retrieval, we use semantic, lexical, and profile-aware scoring: $\text{score}_0(q, e) = \alpha \text{sem}(q, e) + \beta \text{lex}(q, e) + \gamma \text{bias}(q, e)$. Unless otherwise stated, the default weights are $\alpha = 0.60$, $\beta = 0.30$, and $\gamma = 0.10$. Runtime artifacts record retrieval context tokens, context event and block counts, source-turn coverage, turn-level evidence recall, evidence dilution, evidence fragmentation, event-flush tokens, answer-generation tokens, and LLM calls per thousand questions.

H Prompt Templates

Event Grounding Prompt

```
You are the extraction step for an event-centric memory system.

Your task is to convert one flush candidate into one or more local event graph templates.
Every flush candidate in this pipeline contains at least one evidence-grounded event.
Your output must therefore contain at least one event.
The payload contains:
- buffered turns
- lightweight turn sketches
- participant hints
- dominant state keys observed online
- time profile observed online
- correction cues
- a rolling summary

Use candidate.turns as the source of truth.
Use sketches and online fields only as extraction hints.
You must preserve evidence fidelity.
You must not return guessed fields.
You must not omit required fields.

Return one strict JSON object and nothing else.
Do not add markdown.
Do not add code fences.
Do not explain your reasoning.

Return JSON with exactly this top-level shape:
{
  "events": [
    {
      "summary_short": "short event summary",
      "dominant_state_keys": ["entity:attribute=value"],
      "time_profile": ["time cue"],
      "semantic_nodes": [
        {
          "semantic_type": "state|time|entity|relation|constraint",
          "key": "stable semantic key",
          "value": "short value",
          "source_turn_ids": ["turn id"]
        }
      ]
    }
  ],
  "support_blocks": [
    {
      "text": "minimal supporting text span copied or lightly normalized from the turns",
      "semantic_keys": ["matching semantic key"],
      "source_turn_ids": ["turn id"]
    }
  ]
}
```

```

    ]
  }
]
}

```

Core extraction rules:

- EventRoot is the main unit.
- Return the smallest local phase that is still evidence-supported.
- Split only when the candidate truly contains multiple weakly connected phases.
- Preserve the temporal order implied by the candidate.
- Never return an empty events array.
- If the candidate is noisy or underspecified, return exactly one conservative event that covers the whole candidate without inventing details.

When to return one event:

- The turns revolve around one main state line, one local goal, or one local time center.
- Later turns mainly clarify, confirm, refine, or extend the same local phase.

When to split into multiple events:

- A new main time center appears and it is not just detail.
- A new main state-change cluster appears and it cannot be naturally attached to the first one.
- The evidence naturally groups into separate phases with distinct summaries.

Do not split for:

- rephrasing
- short clarifications
- repeated confirmations
- minor detail additions about the same local phase

Field requirements:

For "summary_short":

- Use one short sentence or sentence fragment.
- Keep it retrieval-friendly.
- Mention the main entity, action, and resulting state when possible.
- Keep it under 160 characters when possible.

For "dominant_state_keys":

- Use stable keys in the form "entity:attribute=value".
- Keep only retrieval-critical state keys.
- Prefer 1 to 4 items.
- Do not use vague placeholders.

For "time_profile":

- Keep only concrete time cues or ordering cues that matter for this event.
- Use normalized short phrases grounded in the turns.
- Prefer 0 to 4 items.

For "semantic_nodes":

- Each semantic node must include semantic_type, key, value, and source_turn_ids.
- source_turn_ids must point to one or more turns in the candidate that support the node.
- Prefer 1 to 5 semantic nodes per event.
- Do not create a semantic node unless it helps retrieve or connect a support block.
- Use semantic_type meanings strictly:
 - state: persistent state or updated state
 - time: date, schedule, ordering cue, time anchor
 - entity: person, team, place, object, organization
 - relation: stable action or relation between entities
 - constraint: requirement, condition, limitation, commitment
- Keep keys stable and reusable.
- Include the main entity and attribute in the key when possible.
- Good keys: "melanie:camping_place", "caroline:symbol", "jon:dance_plan".
- Bad keys: "activity", "location", "event_time", "support", "participant".
- Do not output duplicate nodes with slightly different wording.

For "support_blocks":

- Each support block must include text, semantic_keys, and source_turn_ids.
- text must stay close to the source wording.
- source_turn_ids must identify the supporting turns for that block.
- semantic_keys must reference keys that actually appear in semantic_nodes.
- Cover each answer-bearing turn with support blocks.
- Each support block should express exactly one atomic answer-bearing fact.
- Prefer 8 to 35 words per support block.
- Do not copy an entire turn unless the entire turn is one short atomic fact.
- Prefer 1 to 4 high-value support blocks per event.
- Use more only when the candidate contains clearly distinct answer-bearing facts.
- Prefer one source_turn_id per support block.
- Use multiple source_turn_ids only when one fact is explicitly completed across those turns.
- Do not synthesize a support block by merging facts from distant turns.
- Each support block should be minimal but sufficient evidence.

Answer-bearing evidence coverage:

- Create separate support blocks for distinct facts that are likely to answer who, what, when, where, which, whose, and changed-value questions.
- Preserve answer surfaces for names, dates, jobs, schools, cities, relationships, preferences, plans, activities, places, objects, symbols, hobbies, songs, gifts, possessions, health states, emotions, reasons, and outcomes.

- For list-like facts, preserve each item as a separate compact support block when possible.
- If one turn contains several independent facts, use multiple compact support blocks that may cite the same source turn.
- For time-bearing facts, create one block that contains both the time expression and the action or state it anchors.
- Add a separate time node only when it helps retrieval, but the support block must remain answerable by itself.
- Do not spend support blocks on greetings, generic encouragement, pure acknowledgements, repeated paraphrases, or user questions unless they state a factual detail needed by the event.
- A question turn can be evidence only when it states a concrete fact, preference, plan, relationship, or time anchor.
- Do not preserve the question intent itself as evidence.

Evidence discipline:

- Use only information stated or directly supported by the candidate turns.
- Do not invent corrected values, dates, or hidden motives.
- Do not infer unsupported entity relations.
- If something is uncertain, leave it out instead of guessing.
- Uncertainty is never a reason to return zero events.

Correction handling:

- If the candidate contains an explicit correction, represent the corrected state.
- Create a support block for the corrected value.
- Include the superseded value in a separate support block only if it is explicitly stated and useful to understand the correction.
- Do not invent a correction if the turns only show uncertainty.

Hard output constraints:

- Keep all output in English.
- Do not create IDs.
- Do not omit source_turn_ids in semantic_nodes or support_blocks.
- Do not return empty semantic_nodes.
- Do not return empty support_blocks.
- Do not return support_blocks whose semantic_keys are missing from semantic_nodes.
- Do not return support_blocks whose source_turn_ids are not in the candidate.

Answer Generation Prompt

You answer factual questions using only the supplied evidence.

Return exactly one JSON object:

```
{"answer": "...", "reasoning_summary": "...", "cited_evidence_ids": ["E1"], "insufficient_reason": null, "commit_reason": "..."}

```

Rules:

- Use evidence rows as the source of truth. Each row has a short id, optional dates/turns, text, resolved_times, and sometimes a short source excerpt.
- Cite evidence ids, not internal event or block ids.
- If temporal_hints resolve a relative expression for matching evidence, answer with the resolved value, not the relative phrase.
- If an evidence row has resolved_times, use the right side of that mapping instead of copying the relative phrase from text.
- For temporal questions, check temporal_hints before answering from evidence text.
- Ignore answer_candidates that are unresolved relative time phrases when temporal_hints contain a resolved value.
- Use a source excerpt only to clarify the same evidence row; do not treat it as unrelated extra context.
- Prefer direct, specific evidence over broad summaries.
- If evidence conflicts and no row resolves the conflict, answer that the context is insufficient.
- Keep the answer concise and do not add information not supported by evidence.